IN THE CLAIMS

The current claims follow. For claims not marked as amended in this response, any difference in the claims below and the previous state of the claims is unintentional and in the nature of a typographical error.

1-26. (Previously Canceled).

27. (Previously Presented) A controller for allocating call identity values to call connections associated with a switch, said switch capable of handling call connections between calling devices and called devices on a plurality of trunk lines associated with said switch, said controller comprising:

N call application nodes capable of executing a plurality of identity server processes that allocate call identity values to said call connections; and

a load sharing group, selecting one of a first and second identity server processes to allocate a call identity value to a new call connection according to a load distribution algorithm,

wherein said first identity server process comprises a first primary-backup identity server group, comprising,

a first primary identity server application, executing on a first call application node, and a first backup identity server application, associated with said first primary identity server application,

and wherein, responsive to a failure of the first primary identity server application, the first backup identity server application assumes the role of first primary identity server application.

28. (Previously Presented) The controller as set forth in Claim 27, wherein

said first identity server process allocates call identity values having a first contiguous range and

said second identity server process allocates call identity values having a second contiguous range different than said first contiguous range.

29. (Previously Presented) The controller as set forth in Claim 27, wherein said load distribution algorithm selects between said first and second identity server processes in an alternating manner.

30. (Previously Presented) The controller as set forth in Claim 27 wherein said load distribution algorithm selects between said first and second identity server processes according to a current processing load of said first identity server process and a current processing load of said second identity server process.

31. (Previously Presented) The controller as set forth in Claim 30, wherein said load distribution algorithm selects between said first and second identity server processes in order to maintain said current processing load of said first identity server application at a level substantially equal to said current processing load of said second identity server application.

32. (Previously Presented) The controller as set forth in Claim 27, wherein call state information associated with said first primary identity server application is mirrored to said first backup identity server application.

33. (Previously Presented) The controller as set forth in Claim 32, wherein said first backup identity server application is executing on said first call application node.

34. (Previously Presented) The controller as set forth in Claim 32, wherein said first backup identity server application is executing on a second call application node.

35. (Previously Presented) The controller as set forth in Claim 27, wherein

said second identity server process comprises a second primary-backup server group, comprising

a second primary identity server application, executing on a second call application node and a second backup identity server application, associated with said second primary identity server application.

36. (Previously Presented) The controller as set forth in Claim 35, wherein call state information associated with said second primary identity server application is mirrored to said second backup identity server application.

37. (Previously Presented) The controller as set forth in Claim 36, wherein said second backup identity server application is executing on said second call application node.

38. (Previously Presented) The controller as set forth in Claim 36, wherein said second backup identity server application is executing on said first call application node.

39. (Previously Presented) A wireless network comprising:

a plurality of base stations capable of communicating with a plurality of mobile stations in a coverage are of said wireless network; and

a mobile switching center coupled to said plurality of base stations and to a public switched telephone network by a plurality of trunk lines, said mobile switching center comprising a controller for allocating call identity values to call connections associated with a mobile station, said controller comprising:

N call application nodes capable of executing a plurality of identity server processes that allocate call identity values to said call connections, and

a load sharing group, selecting one of a first and second identity server processes to allocate a call identity value to a new call connection according to a load distribution algorithm,

wherein said first identity server process comprises a first primary-backup identity server group, comprising,

a first primary identity server application, executing on a first call application node, and

a first backup identity server application, associated with said first primary identity server application,

and wherein, responsive to a failure of the first primary identity server application, the first backup identity server application assumes the role of first primary identity server application.

40. (Previously Presented) The wireless network as set forth in Claim 39, wherein

said first identity server process allocates call identity values having a first contiguous range and

said second identity server process allocates call identity values having a second contiguous range different than said first contiguous range.

41. (Previously Presented) The wireless network as set forth in Claim 39, wherein said load distribution algorithm selects between said first and second identity server processes in an alternating manner.

42. (Previously Presented) The wireless network as set forth in Claim 39, wherein said load distribution algorithm selects between said first and second identity server processes according to a current processing load of said first identity server process and a current processing load of said second identity server process.

43. (Previously Presented) The controller as set forth in Claim 42, wherein said load distribution algorithm selects between said first and second identity server processes in order to maintain said current processing load of said first identity server application at a level substantially equal to said current processing load of said second identity server application.

44. (Previously Presented) The controller as set forth in Claim 39, wherein call state information associated with said first primary identity server application is mirrored to said first backup identity server application.

45. (Previously Presented) The controller as set forth in Claim 44, wherein said first backup identity server application is executing on said first call application node.

46. (Previously Presented) The controller as set forth in Claim 44, wherein said first backup identity server application is executing on a second call application node.

47. (Previously Presented) The controller as set forth in Claim 39, wherein

said second identity server process comprises a second primary-backup server group, comprising

a second primary identity server application, executing on a second call application node and

a second backup identity server application, associated with said second primary identity server application.

48. (Previously Presented) The controller as set forth in Claim 47, wherein call state information associated with said second primary identity server application is mirrored to said second backup identity server application.

49. (Previously Presented) The controller as set forth in Claim 48, wherein said second backup identity server application is executing on said second call application node.

50. (Previously Presented) The controller as set forth in Claim 48, wherein said second backup identity server application is executing on said first call application node.